**DAY2**

Log  
Volley  
WebService Access  
Custom Classess  
Accessing UI elements in the code

**Permissions**

Beginning in Android 6.0 (API level 23), users grant permissions to apps while the app is running, not when they install the app. This approach streamlines the app install process, since the user does not need to grant permissions when they install or update the app. It also gives the user more control over the app's functionality; for example, a user could choose to give a camera app access to the camera but not to the device location. The user can revoke the permissions at any time, by going to the app's Settings screen.

System permissions are divided into two categories, *normal* and *dangerous:*

* Normal permissions do not directly risk the user's privacy. If your app lists a normal permission in its manifest, the system grants the permission automatically.
* Dangerous permissions can give the app access to the user's confidential data. If your app lists a normal permission in its manifest, the system grants the permission automatically. If you list a dangerous permission, the user has to explicitly give approval to your app.

**Note:** Beginning with Android 6.0 (API level 23), users can revoke permissions from any app at any time, even if the app targets a lower API level. You should test your app to verify that it behaves properly when it's missing a needed permission, regardless of what API level your app targets.

**Check For Permissions**

If your app needs a dangerous permission, you must check whether you have that permission every time you perform an operation that requires that permission. The user is always free to revoke the permission, so even if the app used the camera yesterday, it can't assume it still has that permission today.

To check if you have a permission, call the

[ContextCompat.checkSelfPermission()](https://developer.android.com/reference/android/support/v4/content/ContextCompat.html#checkSelfPermission%2528android.content.Context,%2520java.lang.String%2529) method. For example, this snippet shows how to check if the activity has permission to write to the calendar:

// Assume thisActivity is the current activity

int permissionCheck = ContextCompat.checkSelfPermission(thisActivity,  
Manifest.permission.WRITE\_CALENDAR);

If the app has the permission, the method returns

[PackageManager.PERMISSION\_GRANTED](https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_GRANTED),

and the app can proceed with the operation. If the app does not have the permission, the method returns [PERMISSION\_DENIED](https://developer.android.com/reference/android/content/pm/PackageManager.html#PERMISSION_DENIED), and the app has to explicitly ask the user for permission.

**Request Permissions**

If your app needs a dangerous permission that was listed in the app manifest, it must ask the user to grant the permission. Android provides several methods you can use to request permission. Calling these methods brings up a standard Android dialog, which you cannot customize.

**Shared Preferences**

**SharedPreferences** is an API from **Android** SDK to store and retrieve application preferences. **SharedPreferences** are simply sets of data values that stored persistently.

How to store value:

SharedPreferences pref = getApplicationContext().getSharedPreferences("ids", 0);

SharedPreferences.Editor editor = pref.edit();

editor.putString("regId", gettoken);//get token contains value

editor.commit();

How to getvalue:

SharedPreferences pref = getApplicationContext().getSharedPreferences("ids", 0);

String regId = pref.getString("regId", null);

Log.e(TAG, "Firebase reg id: " + regId);

**Advanced UIDesign**

**Material Design**

For implementing material design we have add this library to app gradle:

implementation **'com.android.support:design:27.1.1'**

Android provides two new widgets for displaying cards and lists with material design.

For Recycler view we have to add :

implementation 'com.android.support:recyclerview-v7:27.1.1'

For Card view we have to add :

implementation 'com.android.support:cardview-v7:27.1.1'

Now lets make new activities and link to dashboard .The code given below can be used to open another activity from the present UI context.

Intent in1 = **new** Intent(**context**, SearchActivity.**class**);  
 **context**.startActivity(in1);

Now open ImageAdaper.Java and uncomment the below block. Here the activity will be triggered while clicking the grid item

**if**(position==0)  
{  
 Intent in1 = **new** Intent(**context**, SearchActivity.**class**);  
 **context**.startActivity(in1);  
  
}

Now open DashboardActivity.Java and uncomment the below block

Here the activity will be triggered while clicking the menu item

**if** (id == R.id.***search***) {  
   
 Intent i=**new** Intent(DashboardActivity.**this**,SearchActivity.**class**);  
 startActivity(i);  
  
}

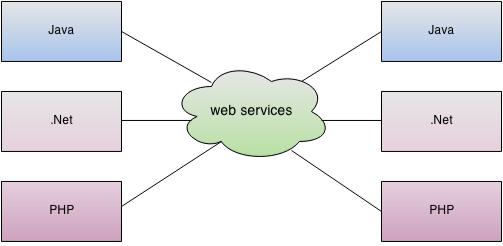
**Introduction to webservice**

**What is Web Service?**

A Web Service can be defined in following ways:

* A client server application or application component for communication.
* A method of communication between two devices over network.
* A software system for interoperable machine to machine communication.
* A collection of standards or protocols for exchanging information between two devices or application.

Let's understand it by the figure given below:



As you can see in the figure, java, .net or PHP applications can communicate with other applications through web service over the network. For example, java application can interact with Java, .Net

and PHP applications. So web service is a language independent way of communication.

**Types of Web Services**

There are mainly two types of web services.

* + SOAP web services.
  + RESTful web services.

**SOAP Web Services**

SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.SOAP is a W3C recommendation for communication between two applications.SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

a. Advantages of Soap Web Services

**WS Security**: SOAP defines its own security known as WS Security.

**Language and Platform independent**: SOAP web services can bewritten in any programming language and executed in any platform.

b. Disadvantages of Soap Web Services

Slow**: SOAP uses XML format that must be parsed to be read. Itdefines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.**

WSDL dependent**:** **SOAP** **uses** **WSDL(Web** **Services** **Description**

**Language) and doesn't have any other mechanism to discover the service.**

**RESTful Web Services**

REST stands for Representational State Transfer.

REST is an architectural style not a protocol.

a. Advantages of RESTful Web Services

**Fast**: RESTful Web Services are fast because there is no strictspecification like SOAP. It consumes less bandwidth and resource.

**Language and Platform independent**: RESTful web services can bewritten in any programming language and executed in any platform.

**I can use SOAP**: RESTful web services can use SOAP web services asthe implementation.

**Permits different data format**: RESTful web service permits differentdata format such as Plain Text, HTML, XML and JSO

**Volley library** can be used for calling webservice within an app. If we are using volley library mention the following line in build.gradle

implementation 'com.android.volley:volley:1.1.0'

Advantages of using volley library:

1. Volley automatically schedule all network requests. It means that Volley will be taking care of all the network requests your app executes for fetching response or image from web.
2. Volley provides transparent disk and memory caching.
3. Volley provides powerful cancellation request API. It means that you can cancel a single request or you can set blocks or scopes of requests to cancel.
4. Volley provides powerful customization abilities.
5. Volley provides Debugging and tracing tools

While calling webservice we have to give certain permissions in Android Manifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="nic.kerala.training"**>  
 **<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />**

**<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />  
 <uses-permission android:name="android.permission.INTERNET" />  
  
 <uses-permission android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES" />  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.READ\_PHONE\_STATE" />**  
 *<!--  
 The ACCESS\_COARSE/FINE\_LOCATION permissions are not required to use  
 Google Maps Android API v2, but you must specify either coarse or fine  
 location permissions for the 'MyLocation' functionality.   
 -->* <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
  
 android:theme="@style/AppTheme"**>  
 <**activity  
 android:name=".Splashactivity"  
 android:theme="@style/AppTheme.NoActionBar"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity  
 android:name=".SqlActivity"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".RegisterActivity"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".NotificationActivity"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".SearchActivity"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".Feedback"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".FileUpload"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".RecyclerviewExample"  
 android:theme="@style/AppTheme.NoActionBar"** />  
 <**activity  
 android:name=".SlidingMenuActivity"  
 android:theme="@style/AppTheme.NoActionBar"** />  
  
 <**service android:name=".MessageReceiver"**>  
 <**intent-filter**>  
 <**action android:name="com.google.firebase.MESSAGING\_EVENT"** />  
 </**intent-filter**>  
 </**service**>  
 <**service android:name=".InstanceIdService"**>  
 <**intent-filter**>  
 <**action android:name="com.google.firebase.INSTANCE\_ID\_EVENT"** />  
 </**intent-filter**>  
 </**service**>  
  
  
  
 <**meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="@string/google\_maps\_key"** />  
  
 <**activity  
 android:name=".MapsActivity"  
 android:label="@string/title\_activity\_maps"** ></**activity**>  
 </**application**>  
  
</**manifest**>

(You may not have all the keys as shown above)

Create following common classes for further development:

*1.Base64Utils*

*It is used for encoding purposes*

**package** in.nic.kerala.training;  
  
**public class** Base64Utils {  
  
 **private static byte**[] *mBase64EncMap*, *mBase64DecMap*;  
  
 */\*\*  
 \* Class initializer. Initializes the Base64 alphabet (specified in  
 \* RFC-2045).  
 \*/* **static** {  
 **byte**[] base64Map = { (**byte**) **'A'**, (**byte**) **'B'**, (**byte**) **'C'**, (**byte**) **'D'**,  
 (**byte**) **'E'**, (**byte**) **'F'**, (**byte**) **'G'**, (**byte**) **'H'**, (**byte**) **'I'**,  
 (**byte**) **'J'**, (**byte**) **'K'**, (**byte**) **'L'**, (**byte**) **'M'**, (**byte**) **'N'**,  
 (**byte**) **'O'**, (**byte**) **'P'**, (**byte**) **'Q'**, (**byte**) **'R'**, (**byte**) **'S'**,  
 (**byte**) **'T'**, (**byte**) **'U'**, (**byte**) **'V'**, (**byte**) **'W'**, (**byte**) **'X'**,  
 (**byte**) **'Y'**, (**byte**) **'Z'**, (**byte**) **'a'**, (**byte**) **'b'**, (**byte**) **'c'**,  
 (**byte**) **'d'**, (**byte**) **'e'**, (**byte**) **'f'**, (**byte**) **'g'**, (**byte**) **'h'**,  
 (**byte**) **'i'**, (**byte**) **'j'**, (**byte**) **'k'**, (**byte**) **'l'**, (**byte**) **'m'**,  
 (**byte**) **'n'**, (**byte**) **'o'**, (**byte**) **'p'**, (**byte**) **'q'**, (**byte**) **'r'**,  
 (**byte**) **'s'**, (**byte**) **'t'**, (**byte**) **'u'**, (**byte**) **'v'**, (**byte**) **'w'**,  
 (**byte**) **'x'**, (**byte**) **'y'**, (**byte**) **'z'**, (**byte**) **'0'**, (**byte**) **'1'**,  
 (**byte**) **'2'**, (**byte**) **'3'**, (**byte**) **'4'**, (**byte**) **'5'**, (**byte**) **'6'**,  
 (**byte**) **'7'**, (**byte**) **'8'**, (**byte**) **'9'**, (**byte**) **'+'**, (**byte**) **'/'** };  
 *mBase64EncMap* = base64Map;  
 *mBase64DecMap* = **new byte**[128];  
 **for** (**int** i = 0; i < *mBase64EncMap*.**length**; i++)  
 *mBase64DecMap*[*mBase64EncMap*[i]] = (**byte**) i;  
 }  
  
 */\*\*  
 \* This class isn't meant to be instantiated.  
 \*/* **private** Base64Utils() {  
 }  
  
 */\*\*  
 \* Encodes the given byte[] using the Base64-encoding, as specified in  
 \* RFC-2045 (Section 6.8).  
 \*   
 \** ***@param aData*** *\* the data to be encoded  
 \** ***@return*** *the Base64-encoded <var>aData</var>  
 \** ***@exception*** *IllegalArgumentException  
 \* if NULL or empty array is passed  
 \*/* **public static** String base64Encode(**byte**[] aData) {  
 **if** ((aData == **null**) || (aData.**length** == 0))  
 **throw new** IllegalArgumentException(  
  
 **"Cannot encode NULL or empty byte array."**);  
  
 **byte** encodedBuf[] = **new byte**[((aData.**length** + 2) / 3) \* 4];  
  
 *// 3-byte to 4-byte conversion* **int** srcIndex, destIndex;  
 **for** (srcIndex = 0, destIndex = 0; srcIndex < aData.**length** - 2; srcIndex += 3) {  
 encodedBuf[destIndex++] =  
  
 *mBase64EncMap*[(aData[srcIndex] >>> 2) & 077];  
 encodedBuf[destIndex++] = *mBase64EncMap*[  
  
 (aData[srcIndex + 1] >>> 4) & 017 | (aData[srcIndex] << 4) & 077];  
 encodedBuf[destIndex++] = *mBase64EncMap*[  
  
 (aData[srcIndex + 2] >>> 6) & 003 | (aData[srcIndex + 1] << 2)  
 & 077];  
 encodedBuf[destIndex++] = *mBase64EncMap*[  
  
 aData[srcIndex + 2] & 077];  
 }  
  
 *// Convert the last 1 or 2 bytes* **if** (srcIndex < aData.**length**) {  
 encodedBuf[destIndex++] =  
  
 *mBase64EncMap*[(aData[srcIndex] >>> 2) & 077];  
 **if** (srcIndex < aData.**length** - 1) {  
 encodedBuf[destIndex++] = *mBase64EncMap*[(aData[srcIndex + 1] >>> 4)  
 & 017 | (aData[srcIndex] << 4) & 077];  
 encodedBuf[destIndex++] =  
  
 *mBase64EncMap*[(aData[srcIndex + 1] << 2) & 077];  
 } **else** {  
 encodedBuf[destIndex++] =  
  
 *mBase64EncMap*[(aData[srcIndex] << 4) & 077];  
 }  
 }  
  
 *// Add padding to the end of encoded data* **while** (destIndex < encodedBuf.**length**) {  
 encodedBuf[destIndex] = (**byte**) **'='**;  
 destIndex++;  
 }  
  
 String result = **new** String(encodedBuf);  
 **return** result;  
 }  
  
 */\*\*  
 \* Decodes the given Base64-encoded data, as specified in RFC-2045 (Section  
 \* 6.8).  
 \*   
 \** ***@param aData*** *\* the Base64-encoded aData.  
 \** ***@return*** *the decoded <var>aData</var>.  
 \** ***@exception*** *IllegalArgumentException  
 \* if NULL or empty data is passed  
 \*/* **public static byte**[] base64Decode(String aData) {  
 **if** ((aData == **null**) || (aData.length() == 0))  
 **throw new** IllegalArgumentException(  
  
 **"Cannot decode NULL or empty string."**);  
  
 **byte**[] data = aData.getBytes();  
  
 *// Skip padding from the end of encoded data* **int** tail = data.**length**;  
 **while** (data[tail - 1] == **'='**)  
 tail--;  
  
 **byte** decodedBuf[] = **new byte**[tail - data.**length** / 4];  
  
 *// ASCII-printable to 0-63 conversion* **for** (**int** i = 0; i < data.**length**; i++)  
 data[i] = *mBase64DecMap*[data[i]];  
  
 *// 4-byte to 3-byte conversion* **int** srcIndex, destIndex;  
 **for** (srcIndex = 0, destIndex = 0; destIndex < decodedBuf.**length** - 2; srcIndex += 4, destIndex += 3) {  
 decodedBuf[destIndex] = (**byte**)  
  
 (((data[srcIndex] << 2) & 255) | ((data[srcIndex + 1] >>> 4) & 003));  
 decodedBuf[destIndex + 1] = (**byte**)  
  
 (((data[srcIndex + 1] << 4) & 255) | ((data[srcIndex + 2] >>> 2) & 017));  
 decodedBuf[destIndex + 2] = (**byte**)  
  
 (((data[srcIndex + 2] << 6) & 255) | (data[srcIndex + 3] & 077));  
 }  
  
 *// Handle last 1 or 2 bytes* **if** (destIndex < decodedBuf.**length**)  
 decodedBuf[destIndex] = (**byte**) (  
  
 ((data[srcIndex] << 2) & 255) | ((data[srcIndex + 1] >>> 4) & 003));  
 **if** (++destIndex < decodedBuf.**length**)  
 decodedBuf[destIndex] = (**byte**)  
  
 (((data[srcIndex + 1] << 4) & 255) | ((data[srcIndex + 2] >>> 2) & 017));  
  
 **return** decodedBuf;  
 }  
  
}

*Create following contents*

*(Note: please include required resource files to rectify errors)*

**layout\_alert.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:background="@drawable/alertstyle"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/txt\_data"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:gravity="center"  
 android:text="ghugh"  
 android:textColor="#8B4513"  
 android:textSize="18dp"** />  
  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="gfshdgfg"  
 android:visibility="invisible"** />  
  
 <**Button  
 android:id="@+id/btnok"  
 android:layout\_width="60dp"  
 android:layout\_height="40dp"  
 android:layout\_gravity="center"  
 android:background="@drawable/activity\_buttonstyle"  
  
 android:text="OK"  
 android:textColor="#ffffff"** />  
  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="gfshdgfg"  
 android:visibility="invisible"** />  
  
  
 </**LinearLayout**>  
</**LinearLayout**>

**layout\_alert\_cust.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="250dp"  
 android:layout\_height="150dp"  
 android:layout\_gravity="center"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/txt\_data"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:gravity="center"  
 android:text="ghugh"  
 android:textColor="#409fe8"  
 android:textSize="18dp"** />  
  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="gfshdgfg"  
 android:visibility="invisible"** />  
  
 <**Button  
 android:id="@+id/btnok"  
 android:layout\_width="60dp"  
 android:layout\_height="40dp"  
 android:layout\_gravity="center"  
 android:background="#409fe8"  
 android:text="OK"  
 android:textColor="#ffffff"** />  
 </**LinearLayout**>  
</**LinearLayout**>

*2.Utils*

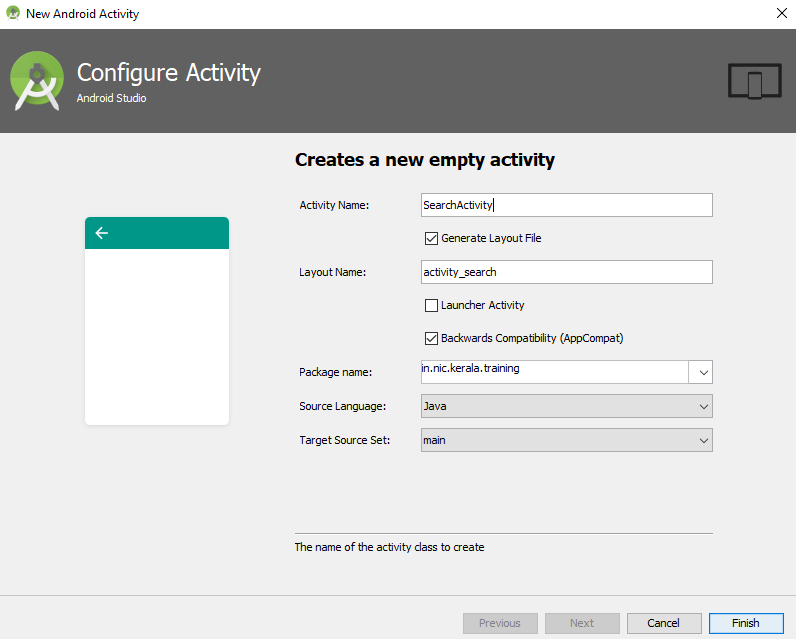
*In this class we have written some common function we are using repeatedly*

**package** in.nic.kerala.training;  
  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.app.Dialog;  
**import** android.content.Context;  
**import** android.net.ConnectivityManager;  
**import** android.net.NetworkInfo;  
**import** android.view.View;  
**import** android.view.Window;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
  
**public class** Utils {  
 **static** Context *context*;  
 **private static** Activity *activity*;  
  
 **private static** AlertDialog *dialog*;  
 **public** Utils(Context con) {  
 *context* = con;  
 *activity* = (Activity) con;  
  
 }  
  
 **public void** alertButton(**final** String data, **final boolean** closeactivity) {  
 *activity*.runOnUiThread(**new** Runnable() {  
 **public void** run() {  
  
 **final** Dialog dialog = **new** Dialog(*activity*);  
 dialog.requestWindowFeature(Window.***FEATURE\_NO\_TITLE***);  
 dialog.setContentView(R.layout.***layout\_alert\_cust***);  
 TextView text = (TextView) dialog.findViewById(R.id.***txt\_data***);  
 text.setText(data);  
  
  
 Button dialogButton = (Button) dialog  
 .findViewById(R.id.***btnok***);  
 dialogButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
  
 **if** (closeactivity == **true**) {  
 dialog.dismiss();  
  
 *activity*.finish();  
 } **else** {  
 dialog.dismiss();  
  
 }  
 }  
 });  
  
 dialog.show();  
  
 }  
 });  
  
 }  
 **public void** alertSingleButton(**final** String data, **final** EditText edt) {  
 *activity*.runOnUiThread(**new** Runnable() {  
 **public void** run() {  
 **if** (*dialog* == **null**) {  
  
 **final** Dialog dialog = **new** Dialog(*activity*);  
 dialog.setContentView(R.layout.***layout\_alert***);  
 dialog.setTitle(**"Message...."**);  
  
 *// set the custom dialog components - text, image and button* TextView text = (TextView) dialog.findViewById(R.id.***txt\_data***);  
 text.setText(data);  
  
  
 Button dialogButton = (Button) dialog  
 .findViewById(R.id.***btnok***);  
 *// if button is clicked, close the custom dialog* dialogButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
  
 **if** (edt != **null**) {  
 dialog.dismiss();  
 edt.requestFocus();  
 } **else** {  
 dialog.dismiss();  
 }  
 }  
 });  
  
 dialog.show();  
  
 }  
 }  
 });  
 }  
  
  
  
  
 **public static boolean** isOnline(Context context) {  
  
 ConnectivityManager connectivity = (ConnectivityManager) context  
 .getSystemService(Context.***CONNECTIVITY\_SERVICE***);  
 **if** (connectivity != **null**) {  
 NetworkInfo[] info = connectivity.getAllNetworkInfo();  
 **if** (info != **null**)  
 **for** (**int** i = 0; i < info.**length**; i++)  
 **if** (info[i].getState() == NetworkInfo.State.***CONNECTED***) {  
 **return true**;  
 }  
 }  
 **return false**;  
 }  
  
  
}

**Access web service**

**Feedback Form**

Create an Activity "SearchActivity".Here we are giving application number and phone number as input and we will get the status of application.



**Modify activity\_search.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
  
 <**include  
 android:id="@+id/head"  
 layout="@layout/activity\_title"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"** />  
  
  
 <**android.support.design.widget.TextInputLayout  
 android:id="@+id/input\_layout\_app"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/head"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"**>  
  
 <**EditText  
 android:id="@+id/input\_app"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Application Number\*"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**android.support.design.widget.TextInputLayout  
 android:id="@+id/input\_layout\_mob"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/input\_layout\_app"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"**>  
  
 <**EditText  
 android:id="@+id/input\_mob"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Mobile Number Given In The Application\*"  
 android:inputType="number"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**Button  
 android:id="@+id/btnsub"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/input\_layout\_mob"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/buttoncustom"  
 android:onClick="onSearchClick"  
 android:text="Submit"  
 android:textColor="#000000"** />  
  
 <**LinearLayout  
 android:id="@+id/details"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btnsub"  
 android:orientation="vertical"  
 android:visibility="invisible"**>  
  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:background="#B2DCF8"  
 android:gravity="center"  
 android:text="Grievance Details"  
 android:textColor="#000000"  
 android:textSize="18dp"** />  
  
 <**TableLayout  
 android:id="@+id/table"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"**>  
  
 <**TableRow  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center|center\_horizontal"**>  
  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Application Number"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
  
 <**TextView  
 android:id="@+id/appnum"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
 </**TableRow**>  
  
 <**TableRow  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center|center\_horizontal"**>  
  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Applicant Name"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
  
 <**TextView  
 android:id="@+id/appname"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
 </**TableRow**>  
  
 <**TableRow  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center|center\_horizontal"**>  
  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
  
 android:layout\_height="wrap\_content"  
 android:text="Status"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
  
 <**TextView  
 android:id="@+id/appsta"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="14dp"  
 android:textStyle="bold"** />  
 </**TableRow**>  
 </**TableLayout**>  
  
 <**TableLayout  
 android:id="@+id/tablelayout\_doc\_list"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="#ffffff"  
  
 android:orientation="vertical"  
 android:shrinkColumns="\*"  
 android:stretchColumns="\*"** >  
  
  
 </**TableLayout**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:id="@+id/in"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="49dp"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/fooo"  
 android:orientation="horizontal"** />  
  
</**RelativeLayout**>

**In drawable put buttoncustom.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape xmlns:android="http://schemas.android.com/apk/res/android"  
android:shape="rectangle"**>  
*<!-- Gradient Bg for listrow -->*<**gradient  
 android:startColor="#87CEFA"  
 android:centerColor="#87CEFA"  
 android:endColor="#87CEFA"  
 android:angle="180"** />  
</**shape**>

**SearchActivity.java**

**package** in.nic.kerala.training;  
  
**import** android.app.AlertDialog;  
**import** android.app.ProgressDialog;  
**import** android.os.Bundle;  
  
**import** android.support.annotation.Nullable;  
**import** android.support.design.widget.TextInputLayout;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.LinearLayout;  
**import** android.widget.TableLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.android.volley.DefaultRetryPolicy;  
**import** com.android.volley.Request;  
**import** com.android.volley.RequestQueue;  
**import** com.android.volley.Response;  
**import** com.android.volley.VolleyError;  
**import** com.android.volley.toolbox.StringRequest;  
**import** com.android.volley.toolbox.Volley;  
  
  
**import** org.json.JSONArray;  
**import** org.json.JSONObject;  
  
**import** java.util.Map;  
  
**public class** SearchActivity **extends** AppCompatActivity {  
 **private** EditText **inputapp**, **inputmob**;  
 **private** TextInputLayout **inputlayoutapp**, **inputlayoutmob**;  
 **private** Button **btnok**;  
 TextView **txt\_nam**, **txt\_num**, **txt\_sta**;  
 Utils **util**;  
 String **appno**, **mobno**;  
 ProgressDialog **pdialog**;  
 **private** TableLayout **tblelayout\_docList**;  
 **int certid**;  
 String **no**;  
 **static** String *url1*;  
 **private** ProgressDialog **DownloadDialog**;  
 **protected** String **fileName** = **""**;  
 RequestQueue **requestQueue**;  
 String **url**;  
 Map<String, String> **params**;  
  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_search***);  
  
 **inputapp** = (EditText) findViewById(R.id.***input\_app***);  
 **inputmob** = (EditText) findViewById(R.id.***input\_mob***);  
  
 **btnok** = (Button) findViewById(R.id.***btnsub***);  
 **txt\_num** = (TextView) findViewById(R.id.***appnum***);  
 **txt\_nam** = (TextView) findViewById(R.id.***appname***);  
 **txt\_sta** = (TextView) findViewById(R.id.***appsta***);  
 View includedLayout = findViewById(R.id.***head***);  
 **tblelayout\_docList** = (TableLayout) findViewById(R.id.***tablelayout\_doc\_list***);  
 **tblelayout\_docList**.setVisibility(View.***GONE***);  
  
 TextView txttitle = (TextView) includedLayout.findViewById(R.id.***txttitile***);  
 txttitle.setText(**"Search"**);  
 **util** = **new** Utils(SearchActivity.**this**);  
  
 }  
  
 **public void** onSearchClick(View view) {  
  
 **appno** = **inputapp**.getText().toString();  
 **mobno** = **inputmob**.getText().toString();  
 **tblelayout\_docList**.setVisibility(View.***GONE***);  
  
 **if** (**inputapp**.getText().toString().equals(**""**)) {  
 **util**.alertButton(**"Please Enter Application number"**, **false**);  
  
 } **else if** (**inputmob**.getText().toString().equals(**""**)) {  
 **util**.alertButton(**"Please Enter Mobile Number"**, **false**);  
 } **else** {  
  
 **if** (Utils.*isOnline*(SearchActivity.**this**)) {  
 **pdialog** = **new** ProgressDialog(SearchActivity.**this**);  
  
 **pdialog** = **new** ProgressDialog(SearchActivity.**this**);  
 **pdialog**.setMessage(**"Loading..."**);  
 **pdialog**.setCancelable(**false**);  
 **pdialog**.show();  
  
  
 LoadService();  
  
 } **else** {  
 **pdialog**.dismiss();  
  
 Toast.*makeText*(getApplicationContext(),  
 **"No Internet Connection!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 }  
  
 **private void** LoadService() {  
  
  
 **url** = **"http://103.251.43.122/pgmobapp/service/mobileapp/getstatus?mobileNo="** + **mobno** + **"&applNo="** + **appno**;  
 Log.*i*(**"URL"**,**url**);  
 **requestQueue** = Volley.*newRequestQueue*(**this**);  
 **final** StringRequest jsonObjectRequest = **new** StringRequest(Request.Method.***GET***, **url**, **new** Response.Listener<String>() {  
  
 **public void** onResponse(String response) {  
 **try** {  
 **pdialog**.dismiss();  
 JSONObject json = **new** JSONObject(response);  
  
 **if** (json.has(**"Status Response"**)) {  
 JSONObject object1 = json.getJSONObject(**"Status Response"**);  
 JSONArray object11 = object1.getJSONArray(**"Status"**);  
  
 **if** (!(object11.equals(**null**) || object11.equals(**""**))) {  
  
 **no** = (String) object11.getJSONObject(0).get(**"Application No"**);  
 String apname = (String) object11.getJSONObject(1).get(**"Applicant Name"**);  
 String stat = (String) object11.getJSONObject(2).get(**"Status"**);  
 **certid**=object11.getJSONObject(3).getInt(**"CertId"**);  
  
 System.***out***.println(**"number is"** + **no**);  
  
  
 LinearLayout table = (LinearLayout) findViewById(R.id.***details***);  
 table.setVisibility(View.***VISIBLE***);  
 **txt\_nam**.setText(**":"** + apname);  
 **txt\_sta**.setText(**":"** + stat);  
 **txt\_num**.setText(**":"** + **no**);  
 **inputapp**.setText(**""**);  
 **inputmob**.setText(**""**);  
  
 **if**(stat.equals(**"Redressed/Disposed"**)){  
 **tblelayout\_docList**.setVisibility(View.***VISIBLE***);  
  
 }  
  
 }  
 } **else** {  
  
 JSONObject st = json.getJSONObject(**"Result"**);  
 String re = (String) st.get(**"error"**);  
 AlertDialog.Builder alert = **new** AlertDialog.Builder(SearchActivity.**this**);  
 alert.setTitle(**"Message"**);  
 alert.setMessage(re);  
 alert.setPositiveButton(**"OK"**, **null**);  
 alert.show();  
 }  
 }  
  
 **catch** (Exception e) {  
  
 **pdialog**.dismiss();  
 e.printStackTrace();  
 }  
 }  
 }, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError e) {  
 e.printStackTrace();  
 }  
 });  
  
 jsonObjectRequest.setRetryPolicy(**new** DefaultRetryPolicy(10000,  
 DefaultRetryPolicy.***DEFAULT\_MAX\_RETRIES***,  
 DefaultRetryPolicy.***DEFAULT\_BACKOFF\_MULT***));  
 **requestQueue**.getCache().clear();  
 **requestQueue**.add(jsonObjectRequest);  
 }  
  
  
}

**TASK1  
Now Let us do another Demo with WebService. Lets us create a feedback form for a submitted application. Now uncomment the corresponding codes in ImageAdapter and Dashboard Activity to enable the proper navigation.**

Intent i=**new** Intent(DashboardActivity.**this**, FeedbackActivity.**class**);  
startActivity(i);

Above code is used for opening the FeedbackActivity

**You may try to create FeedBackActivity yourself.**

Method for checking the existing Feedback status

**Web service URL details**

**http://103.251.43.122/pgmobapp/service/mobileapp/VerifyPgFeedback**

**Method** POST

Parameters  
**apNo**,**mbNo**

Method for submitting a new Feedback

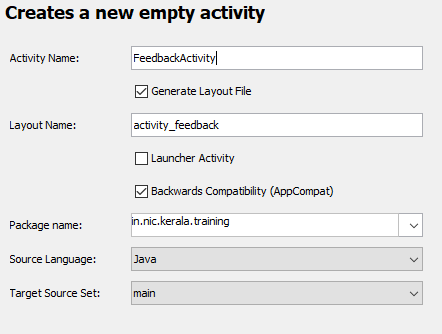
**Web service URL details**

**http://103.251.43.122/pgmobapp/service/mobileapp/savePgFeedback**

**Method** POST

Parameters  
**apNo**, **Feedback**

**Create a new Empty Activity FeedBackActivity**

****

**Open FeedbackActivity.java and paste following code**

**package** in.nic.kerala.training;  
  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.View;  
**import** android.view.inputmethod.InputMethodManager;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.LinearLayout;  
**import** android.widget.Toast;  
  
**import** com.android.volley.DefaultRetryPolicy;  
**import** com.android.volley.Request;  
**import** com.android.volley.RequestQueue;  
**import** com.android.volley.Response;  
**import** com.android.volley.VolleyError;  
**import** com.android.volley.toolbox.StringRequest;  
**import** com.android.volley.toolbox.Volley;  
  
**import** java.net.URLEncoder;  
**import** java.util.HashMap;  
**import** java.util.Map;  
  
  
**public class** FeedbackActivity **extends** AppCompatActivity {  
 EditText **edt\_mobno**, **edt\_appno**, **edt\_feed**;  
 Button **btn\_veri**, **btn\_feed**;  
 Utils **util**;  
 **private** ProgressDialog **pDialog** = **null**;  
 LinearLayout **line**;  
  
 RequestQueue **requestQueue**;  
 String **url**;  
 Map<String, String> **params**;  
 String **feed**;  
 String **apNO**;  
 String **mobno**;  
  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_feedback***);  
 **btn\_veri** = (Button) findViewById(R.id.***btnsub***);  
 **edt\_appno** = (EditText) findViewById(R.id.***input\_app***);  
 **edt\_mobno** = (EditText) findViewById(R.id.***input\_mob***);  
 **edt\_feed** = (EditText) findViewById(R.id.***input\_feed***);  
 **btn\_feed** = (Button) findViewById(R.id.***btnfeed***);  
 **util** = **new** Utils(FeedbackActivity.**this**);  
  
  
 **btn\_veri**.setOnClickListener(**new** View.OnClickListener() {  
  
  
 @Override  
 **public void** onClick(View v) {  
  
 hidekeyboard();  
 **if** (**edt\_appno**.getText().toString().equals(**" "**) || **edt\_mobno**.getText().toString().equals(**" "**)) {  
 **util**.alertSingleButton(**"Please Enter the details"**, **null**);  
 } **else** {  
  
 **apNO** = **edt\_appno**.getText().toString().trim();  
 **mobno** = **edt\_mobno**.getText().toString().trim();  
  
  
 **if** (Utils.*isOnline*(FeedbackActivity.**this**)) {  
  
  
 **pDialog** = **new** ProgressDialog(FeedbackActivity.**this**);  
 **pDialog**.setMessage(**"Loading..."**);  
 **pDialog**.setCancelable(**false**);  
 **pDialog**.show();  
 verify();  
  
  
 } **else** {  
  
 Toast.*makeText*(getApplicationContext(),  
 **"No Internet Connection!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
 }  
  
  
 });  
  
  
 **btn\_feed**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 hidekeyboard();  
  
 **if** (**edt\_feed**.getText().toString().equals(**" "**)) {  
 **util**.alertSingleButton(**"Please Enter feedback"**, **null**);  
 } **else** {  
 **feed** = **edt\_feed**.getText().toString();  
 **apNO** = **edt\_appno**.getText().toString().trim();  
  
  
 **try** {  
  
  
 **if** (Utils.*isOnline*(FeedbackActivity.**this**)) {  
  
  
 **pDialog** = **new** ProgressDialog(FeedbackActivity.**this**);  
 **pDialog**.setMessage(**"Loading..."**);  
 **pDialog**.setCancelable(**false**);  
 **pDialog**.show();  
  
 save();  
  
  
 } **else** {  
  
 Toast.*makeText*(getApplicationContext(),  
 **"No Internet Connection!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 });  
  
  
 }  
  
 **private void** hidekeyboard() {  
  
 InputMethodManager imm = (InputMethodManager) getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  
 imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),  
 InputMethodManager.***RESULT\_UNCHANGED\_SHOWN***);  
 }  
  
 **private void** save() {  
  
 **url** = **"http://103.251.43.122/pgmobapp/service/mobileapp/savePgFeedback"**;  
  
 **requestQueue** = Volley.*newRequestQueue*(**this**);  
  
 **final** StringRequest jsonObjectRequest = **new** StringRequest(Request.Method.***POST***, **url**, **new** Response.Listener<String>() {  
  
 **public void** onResponse(String response) {  
 **try** {  
  
 String Output = (response.trim() == **null**) ? **"submitted"** : response.trim();  
 **if** (Output.equals(**"Feedback submitted Successfully"**)) {  
 **pDialog**.dismiss();  
 **util**.alertButton(Output, **true**);  
  
 } **else** {  
 **pDialog**.dismiss();  
 **util**.alertButton(Output, **true**);  
  
 }  
 } **catch** (Exception e) {  
 **pDialog**.dismiss();  
 e.printStackTrace();  
 }  
 }  
 }, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError e) {  
 e.printStackTrace();  
 }  
 }) {  
 @Override  
  
 **public** Map<String, String> getParams() {  
 **try** {  
 **params** = **new** HashMap<>();  
 **params**.put(**"apNo"**, URLEncoder  
 .*encode*(**apNO**, **"UTF-8"**));  
 **params**.put(**"Feedback"**, **feed**);  
  
 } **catch** (Exception e) {  
  
 }  
 **return params**;  
 }  
 };  
  
 jsonObjectRequest.setRetryPolicy(**new** DefaultRetryPolicy(10000,  
 DefaultRetryPolicy.***DEFAULT\_MAX\_RETRIES***,  
 DefaultRetryPolicy.***DEFAULT\_BACKOFF\_MULT***));  
 **requestQueue**.getCache().clear();  
 **requestQueue**.add(jsonObjectRequest);  
 }  
  
 **private void** verify() {  
  
 **url** = **"http://103.251.43.122/pgmobapp/service/mobileapp/VerifyPgFeedback"**;  
 **requestQueue** = Volley.*newRequestQueue*(**this**);  
  
 **final** StringRequest jsonObjectRequest = **new** StringRequest(Request.Method.***POST***, **url**, **new** Response.Listener<String>() {  
  
 **public void** onResponse(String response) {  
 **try** {  
 String output = (response.trim() == **null**) ? **"Submit"** : response.trim();  
 System.***out***.println(**"output"** + output);  
 **if** (output.equals(**"Success"**)) {  
 **pDialog**.dismiss();  
 **util**.alertButton(**"Now You Can Enter Your Feedback"**, **false**);  
  
 **btn\_veri**.setEnabled(**false**);  
  
 **edt\_appno**.setEnabled(**false**);  
 **edt\_mobno**.setEnabled(**false**);  
 **btn\_feed**.setEnabled(**true**);  
 *// line.setVisibility(View.VISIBLE);* } **else** {  
 **pDialog**.dismiss();  
 **util**.alertButton(output, **true**);  
  
 }  
 } **catch** (Exception e) {  
 **pDialog**.dismiss();  
 e.printStackTrace();  
 **util**.alertButton(**"Connection Error"**,**true**);  
 }  
 }  
 }, **new** Response.ErrorListener() {  
 @Override  
 **public void** onErrorResponse(VolleyError e) {  
 **pDialog**.dismiss();  
 **util**.alertButton(**"Connection Error"**,**true**);  
 e.printStackTrace();  
 }  
 }) {  
 @Override  
  
 **public** Map<String, String> getParams() {  
 **try** {  
 **params** = **new** HashMap<>();  
 **params**.put(**"apNo"**, URLEncoder.*encode*(**apNO**, **"UTF-8"**));  
 **params**.put(**"mbNo"**, URLEncoder.*encode*(**mobno**, **"UTF-8"**));  
  
 } **catch** (Exception e) {  
  
 }  
 **return params**;  
 }  
 };  
  
 jsonObjectRequest.setRetryPolicy(**new** DefaultRetryPolicy(10000,  
 DefaultRetryPolicy.***DEFAULT\_MAX\_RETRIES***,  
 DefaultRetryPolicy.***DEFAULT\_BACKOFF\_MULT***));  
 **requestQueue**.getCache().clear();  
 **requestQueue**.add(jsonObjectRequest);  
 }  
  
  
}

**Open Activity\_feedback.xml**

*<?***xml version="1.0" encoding="utf-8"** *?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**include  
  
 android:id="@+id/head"  
 layout="@layout/activity\_title"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"** />  
  
  
 <**android.support.design.widget.TextInputLayout  
 android:id="@+id/input\_layout\_app"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/head"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"**>  
  
 <**EditText  
 android:id="@+id/input\_app"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Application Number\*"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**android.support.design.widget.TextInputLayout  
 android:id="@+id/input\_layout\_mob"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/input\_layout\_app"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"**>  
  
 <**EditText  
 android:id="@+id/input\_mob"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Mobile Number Given In The Application\*"  
 android:inputType="number"** />  
  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**Button  
 android:id="@+id/btnsub"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:layout\_below="@+id/input\_layout\_mob"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/buttoncustom"  
 android:text="Validate"  
 android:textColor="#000000"** />  
  
 <**android.support.design.widget.TextInputLayout  
 android:id="@+id/input\_layout\_feed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btnsub"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"**>  
  
 <**EditText  
 android:id="@+id/input\_feed"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter Your Feedback\*"** />  
 </**android.support.design.widget.TextInputLayout**>  
  
 <**Button  
 android:id="@+id/btnfeed"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/input\_layout\_feed"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/buttoncustom"  
 android:enabled="false"  
 android:text="Submit"  
 android:textColor="#000000"** />  
  
 <**LinearLayout  
 android:id="@+id/in"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="49dp"  
 android:layout\_alignParentBottom="true"  
 android:background="@drawable/fooo"  
 android:orientation="horizontal"** />  
  
  
</**RelativeLayout**>